

## Adatbázisrendszerek I. - Gyakorlat

### File szintű perzisztencia a C nyelvben, Bináris fa

Készítsen egy mappát a **Desktop**-ra – pl.: **XY\_neptunkod** – ebbe mentse el a feladatokat (XY – mindenkinek a monogramja).

Az elkészült feladatokat csomagolja be és töltse fel a Classroom rendszerbe: **XY\_neptunkod**

#### 1. feladat

Építsen fel egy B-fát az alábbi elemekből, melyek beépülési sorrendje adott.

A fa fokszáma 4, és a beszúrandó elemek listája: 6, 12, 9, 2, 5, 4, 15, 20, 1, 3, 10, 14, 17, 16, 21, 25, 24.

Mentés: *BL\_1.1.pdf*

#### 2. feladat

A programozás feladatokat CodeBlocks programmal készítsék el. A projekt neve: *XY\_neptunkod*

A szabvány billentyűzetről olvasson be sorokat, egészen a végjelig (#). A beolvasott sorokat írja ki egy szövegfile-ba (*munka.txt*). A szövegfile nevét a bevitel első sorában adja meg. Az így létrehozott, lezárt állományt utána nyissa meg és írja vissza a lementett szöveget nagybetűs formában.

Útmutató

```
#include <stdio.h>
#include <stdlib.h>
void pelda_1();

int main()
{
    //printf("Hello world!\n");
    pelda_1();
    return 0;
}

void pelda_1() {
    FILE *fp;
    char ch;
    char fnev[50];
    printf("Filenev: ");
    scanf("%s", fnev);
    fp = fopen(fnev, "w");
    printf("Üzenet ");
    while( (ch = getchar()) != '#' ) {
        putc(ch, fp);
    }
    fclose(fp);

    fp = fopen(fnev, "r");
    while ( (ch = getc(fp)) != EOF ) {
        printf("%c",ch);
    }
    fclose(fp);
    return 0;
}
```

### 3. feladat

Végezze el egy fájl tartalmának másolását egy másik fájlba (*fp1*, *fp2*;). A file neveket másoló függvényt hívó részben kell beolvasni.

Útmutató:

```
void pelda_2(char *fnev1, char *fnev2)
{
    FILE *fp1, *fp2;
    char ch;
    int pos;

    if ((fp1 = fopen(fnev1, "r")) == NULL)
    {
        printf("\nNem lehet megnyitni a fajt. ");
        return;
    }
    else
    {
        printf("\nFajl megnyitva masolasra...\n ");
    }
    fp2 = fopen(fnev2, "w");
    fseek(fp1, 0L, SEEK_END); // File pointer a fajl vegere
    pos = ftell(fp1);
    fseek(fp1, 0L, SEEK_SET); // File pointer a fajl elejere
    while (pos--)
    {
        ch = fgetc(fp1); // Karakterenkent masolja a fajl tartalmat
        fputc(ch, fp2);
    }
    fclose(fp1);
    fclose(fp2);
}
```

### 4. feladat

Tároljon le auto (rendszám, típus, ár) rekordokat egymás után egy bináris állományban, majd készítsen függvényt az i. rekord visszaolvasására.

### 5. feladat

Készítsen programot, mely fel tud vinni személyeket (azonosító és név) bináris fájlba. Készítsen függvényt a) új rekordot létrehozatalára, b) létező rekord törlésére c) létező rekord módosítására

Útmutató:

- használjon bináris file-t
- használjon struct szerkezetet
- használja az `fseek()`, `ftell()` függvényeket
- törlésnél másolja át a maradó részt

## 6. feladat

Az autókat tároló adatfile-ban végezze el az alábbi lekérdezési műveleteket:

- Számítsa ki a fájlban eltárolt autók átlagárát.
- Kérdezze le az eltárolt piros autók darabszámát.
- Keresse meg a legdrágább autót a fájlban.

Útmutató:

- olvassa át az állományt rekordonként
- a szükséges adatokat emelje ki a rekordból
- végezze ez a szükséges számításokat memória változók segítségével.